

УДК 519.688:519.6:514.752

ЯЗЫК ОПИСАНИЯ ТВЕРДОТЕЛЬНЫХ ГЕОМЕТРИЧЕСКИХ МОДЕЛЕЙ FORTU-F

Аль-Омари М. А. В., аспирант

*Запорожский национальный университет,
ул. Жуковского, 66, г. Запорожье, 69600, Украина*

moh.alo1984@mail.ru

Предложен проблемно-ориентированный язык спецификаций (формального описания моделей) FORTU-F, позволяющий полно, непротиворечиво и однозначно описывать математические модели двух- и трехмерных геометрических областей произвольной формы. Для автоматизации использования предложенного языка реализовано интеллектуальное программное средство QNG аналитического конструирования геометрических моделей инженерных конструкций. Приведены примеры описания с его помощью реальных геометрических объектов.

Ключевые слова: математическая модель, геометрический объект, язык спецификаций, автоматизация, интеллектуальное программное средство, FORTU-F.

МОВА ОПИСУ ТВЕРДОТІЛЬНИХ ГЕОМЕТРИЧНИХ МОДЕЛЕЙ FORTU-F

Аль-Омарі М. А. В., аспірант

*Запорізький національний університет,
вул. Жуковського, 66, м. Запоріжжя, 69600, Україна*

moh.alo1984@mail.ru

Запропоновано проблемно-орієнтовану мову специфікацій (формального опису моделей) FORTU-F, яка дозволяє повністю, несуперечливо і однозначно описувати математичні моделі дво- і тривимірних геометричних областей довільної форми. Для автоматизації використання запропонованої мови реалізоване інтелектуальне програмне забезпечення QNG для аналітичного конструювання геометричних моделей інженерних конструкцій. Наведені приклади опису з її допомогою реальних геометричних об'єктів.

Ключові слова: математична модель, геометричний об'єкт, мова специфікацій, автоматизація, інтелектуальне програмне забезпечення, FORTU-F.

LANGUAGE OF SOLID MODELLING FORTU-F

Al-Omari M. A. V., postgraduate

*Zaporizhzhya National University,
Zhukovsky str., 66, Zaporizhzhya, 69600, Ukraine*

moh.alo1984@mail.ru

At present, the analysis of complex processes, objects and phenomena of the real world is virtually impossible without the use of computer simulation. This is due to the fact that the construction and study of mathematical models, based on the cost of wasted resources, are more preferred than the study of the real object (its physical or natural models). The most commonly used in practice, numerical methods, such as method of boundary and finite elements, based on the idea of a transition from the continuous task to some of its discrete analogue, which ultimately requires the establishment of reliable and valid methods for constructing discrete geometrical models of real technical objects complex shape. To automate the geometric modeling of real engineering structures, you must first develop a formal way to describe mathematical relationships and formulas describing the surface of a geometric object suitable for further computer processing. The most effective way to solve this problem is to create a problem-oriented models of formal description languages (specifications), allowing complete, consistent and clearly describe the geometry of objects of arbitrary shape. An example of implementation of such language is FORTU-3. Its main drawback is the inability to describe the R-transactions. Therefore, in this paper, we propose the development of a language, called FORTU-F. Description of a geometric object through language FORTU-F specification consists of the following components (sections): description of the variables and constants; descriptions of functions; describe geometric primitives

support; describing the outcome of a geometric object. The proposed problem-oriented specification language (formal description of models) FORTU-F enables complete, consistent and clearly describe the mathematical models of two- and three-dimensional geometric areas of arbitrary shape. To automate the use of this language is implemented intelligent software tool QNG analytical design of geometric models of engineering structures. The examples describe with the help of real geometric objects.

Key words: mathematical model, geometric object, specification language, automation, intelligent software tool, FORTU-F.

ВВЕДЕНИЕ

В настоящее время анализ сложных процессов, объектов и явлений реального мира практически невозможен без использования компьютерного моделирования. Это обусловлено тем фактом, что построение и изучение математических моделей, исходя из стоимости затраченных ресурсов, являются более предпочтительными, чем изучение реального объекта (его физических или натуральных моделей).

Наиболее часто применяемые на практике численные методы, такие, например, как метод граничных или конечных элементов, базируются на идее перехода от непрерывной задачи к ее некоторому дискретному аналогу, что в конечном итоге требует создания надежных и научно обоснованных методов построения дискретных геометрических моделей реальных технических объектов сложной формы. Для автоматизации геометрического моделирования реальных инженерных конструкций необходимо в первую очередь разработать формальный способ описания математических соотношений и формул, описывающих поверхность геометрического объекта, пригодный для последующей компьютерной обработки. Наиболее эффективным способом решения данной задачи является создание проблемно-ориентированных формальных языков описания моделей (спецификаций) [1], позволяющих полно, непротиворечиво и однозначно описывать геометрию объектов произвольной формы [2-4].

Для адекватного описания математических моделей двух- и трехмерных геометрических объектов сложной формы входной язык описания должен:

- 1) позволять описывать все необходимые математические соотношения для описания моделей плоских и пространственных геометрических объектов произвольной формы;
- 2) позволять задавать управляющие параметры, определяющие точность построения твердотельной модели и дискретизации на конечные элементы заданного типа;
- 3) обладать необходимыми встроенными операторами и операциями для реализации булевых геометрических операторов сложения, вычитания и отрицания;
- 4) поддерживать работу с R-функциями и предикатными формулами;
- 5) поддерживать возможность управления процессом визуализации твердотельной и дискретной модели исходного ГО;
- 6) быть полным и непротиворечивым в пределах своего назначения.

Примером реализации подобного языка является FORTU-3 [3, 5]. Его главным недостатком является невозможность описания R-операций. Поэтому в данной работе предлагается развитие данного языка, получившее название FORTU-F. Описание геометрического объекта с помощью языка спецификаций FORTU-F состоит из следующих частей (секций):

- 1) описания переменных и констант;
- 2) описания функций;
- 3) описания вспомогательных геометрических примитивов;
- 4) описания итогового геометрического объекта.

ОСНОВНЫЕ СИМВОЛЫ ЯЗЫКА

Формальное задание проблемно-ориентированного языка должно представляться описанием его синтаксиса и семантики. Синтаксисом языка называется множество правил, описывающих структуру предложений языка и порядок следования его лексем. Семантикой называется множество правил интерпретации смысла предложения в языке. Совокупность правил синтаксиса формального языка образует его формальную грамматику.

Для однозначного описания синтаксиса и семантики языка FORTU-F можно воспользоваться теорией описания формальной грамматики Н. Хомского [6] и Бэкуса-Наура [7].

Согласно определению Хомского, формальная грамматика представляет собой четверку:

$$G = \{N, T, P, a\}, \quad (1)$$

где N – множество нетерминальных символов языка; T – множество терминальных символов языка; P – множество правил подстановки (все правила состоят из левой и правой части, которые соединяются между собой знаком секвенции « \rightarrow »); все части представляют собой цепочки нетерминальных и/или терминальных символов; правая часть правила определяет цепочку символов, которая может замещать цепочку из левой части); a – аксиома грамматики (множество нетерминальных символов, с которых начинается порождение или распознавание любого предложения языка).

Для описания синтаксиса формальных языков очень удобно использовать Бэкуса-Наура форму (БНФ), которая позволяет одни синтаксические категории последовательно представить через другие категории. Развитие БНФ привело к появлению расширенной формы Бэкуса-Наура (РБНФ) [8], которая отличается от БНФ расширенными возможностями описания сложных языковых конструкций.

Формальное описание языка спецификаций FORTU-F с помощью РБНФ выглядит следующим образом:

буква = «A» | «B» | «C» | «D» | «E» | «F» | «G» | «H» | «I» | «J» | «K» | «L» | «M» | «N»
| «O» | «P» | «Q» | «R» | «S» | «T» | «U» | «V» | «W» | «X» | «Y» | «Z» | «a» | «b» | «c» |
«d» | «e» | «f» | «g» | «h» | «i» | «j» | «k» | «l» | «m» | «n» | «o» | «p» | «q» | «r» | «s» |
«t» | «u» | «v» | «w» | «x» | «y» | «z» | «_»

цифра = «0» | «1» | «2» | «3» | «4» | «5» | «6» | «7» | «8» | «9»

знак = «-» | «+»

разделитель = «+» | «-» | «*» | «/» | «^» | «(» | «)» | «,» | «=» |
зарезервированное_слово

зарезервированное_слово = «abs» | «acos» | «and» | «argument» | «asin» | «atan» |
«atan2» | «begin» | «cutcone» | «cos» | «cosh» | «cylinder» | «ellipsoid» | «end» | «exp» |
«not» | «object» | «or» | «cuboid» | «plane» | «result» | «function» | «sin» | «sinh» |
«sphere» | «tan» | «tanh» | «variable»

идентификатор = буква { буква | цифра }

число_без_знака = целое_без_знака | вещественное_без_знака

число_со_знаком = [знак] число_без_знака

целое_без_знака = последовательность_цифр

последовательность_цифр = цифра {цифра}

вещественное_без_знака = целое_без_знака «.» дробная_часть [«E» порядок] |
целое_без_знака [«E» порядок]

дробная_часть = последовательность_цифр

порядок = целое_со_знаком

целое_со_знаком = [знак] целое_без_знака

комментарий = «!» [ASCII_последовательность]

ASCII_последовательность = пусто | ASCII_символ | ASCII_последовательность
ASCII_символ

пусто =

Здесь термин «идентификатор» обозначает различные названия, используемые в конструкциях языка FORTU-F. Имена идентификаторов могут быть произвольными за исключением того, что не допускается совпадение имен идентификаторов с зарезервированными словами языка. Смысл большинства зарезервированных слов будет пояснен ниже. Понятие «ASCII_символ» формально не расписывается (в терминах РБНФ), под ним понимается любой из 256 стандартных символов таблицы ASCII [48].

ТИПЫ ДАННЫХ И ПРАВИЛА ИХ ОБРАЗОВАНИЯ

Язык спецификаций FORTU-F оперирует двумя типами данных: числами (variable) и функциями (function). Другими словами, все типы данных языка являются формулами (число или константа – простейший вид формулы). Числа в FORTU-F бывают двух типов: целые и вещественные с плавающей точкой. Разрядность хранения чисел определяется текущей компьютерной платформой, в которой реализован транслятор языка FORTU-F. Функциональный тип данных предназначен для хранения формульных выражений, значения которых определяются в зависимости от их аргументов, которые в свою очередь являются числами или функциями (argument, variable или function).

С помощью РБНФ формально тип данных в языке FORTU-F определяется следующим образом:

тип_данных = числовой_тип_данных | функциональный_тип_данных

числовой_тип_данных = число_без_знака | число_со_знаком

функциональный_тип_данных = выражение

Понятие выражения будет рассмотрено ниже. Преобразование типов в языке FORTU-F осуществляется по следующему правилу: в арифметических выражениях, содержащих операнды разных типов, происходит их автоматическое преобразование к наиболее старшему типу. Старшинство типов определяется так (снизу-вверх): целое число – вещественное число – функция.

Переменные в FORTU-F описываются (декларируются) следующим образом:

описание_переменных = тип список_переменных

тип = «argument» | «variable» | «function»

список_переменных = описание_переменной | список_переменных «,»
описание_переменной

описание_переменной = переменная [«=» выражение]

переменная = идентификатор

Тип данных «argument» предназначен для декларации координатных переменных (значения которых при вычислениях рассматриваются как координаты) и размерности геометрической модели. Например, выражение языка

ARGUMENT x, y

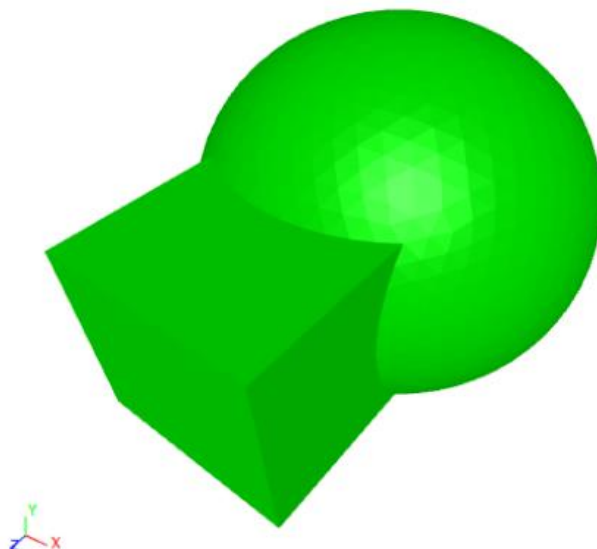


Рис. 1. Область, образованная объединением сферы и прямоугольного параллелепипеда

СТРУКТУРА ОПИСАНИЯ МОДЕЛИ

Структуру описания модели геометрического объекта на языке спецификации FORTU-F можно формально описать следующим образом:

```

FORTU-F_спецификация = блок { блок }
Блок = «object» имя_объекта [«(»[список_параметров]«)»] «begin»
декларативная_часть инструктивная_часть результирующая_часть «end»
имя_объекта = идентификатор
декларативная_часть = описание_переменных { описание_переменных }
инструктивная_часть = { оператор_присваивания }
оператор_присваивания = идентификатор «=» выражение
результирующая_часть = имя_объекта «=» выражение

```

Из приведенного описания следует, что геометрический объект задается с помощью секции «object», которая состоит из трех частей:

- 1) блока деклараций – описания используемых переменных (в т. ч. координатных) и функций;
- 2) блока инструкций, содержащих определения всех ранее задекларированных переменных (этот блок может отсутствовать);
- 3) результирующего блока, задающего итоговую формулу, реализующую математическую модель геометрического объекта.

Следует отметить, что первый блок должен содержать минимум один элемент – секцию «argument», определяющую идентификаторы координат, если геометрическая область описывается в явном виде с использованием, например, R-функций. В случае же, когда используются встроенные геометрические примитивы (будут описаны ниже) или же ранее описанные геометрические объекты, первый блок может быть пустым. Второй блок может отсутствовать, если все необходимые выражения заданы при начальной инициализации переменных или в результирующем выражении. Третий блок, представленный единственным оператором присваивания, определяющим итоговое выражение модели геометрического объекта, также является обязательным.

Таким образом, математическая модель геометрического объекта, описанная с помощью языка спецификации FORTU-F, по сути является непустой совокупностью модулей

(«object»), каждый из которых содержит описание некоторой геометрической фигуры (или ее части). Важно отметить, что описание модуля является параметрическим, т. е. при его декларации имеется возможность задания параметров, которые позволяют обеспечить массовость использования объекта в последующем. Например, описание параметрического объекта «сфера», центр и радиус которой заданы как параметры, может выглядеть так:

OBJECT _sphere_(x0, y0, z0, R)

BEGIN

ARGUMENT x, y, z

$$_sphere_ = R^2 - (x - x0)^2 - (y - y0)^2 - (z - z0)^2$$

END

Здесь в имени объекта используются символы подчеркивания, поскольку, как уже отмечалось выше, не допускается использование имен идентификаторов, совпадающих с зарезервированными словами языка (sphere – название встроенной в FORTU-F функции).

Следует отметить, что описанные объекты в языке спецификаций FORTU-F могут повторно использоваться, в частности для композиции в более сложные объекты. Например, объединение двух сфер (рис. 2) может быть описано как их композиция следующим образом (с учетом ранее приведенного примера):

OBJECT result

BEGIN

result = _sphere_(0, 0, 0, 1) **OR** _sphere_(0, 0, 1.5, 1.5)

END

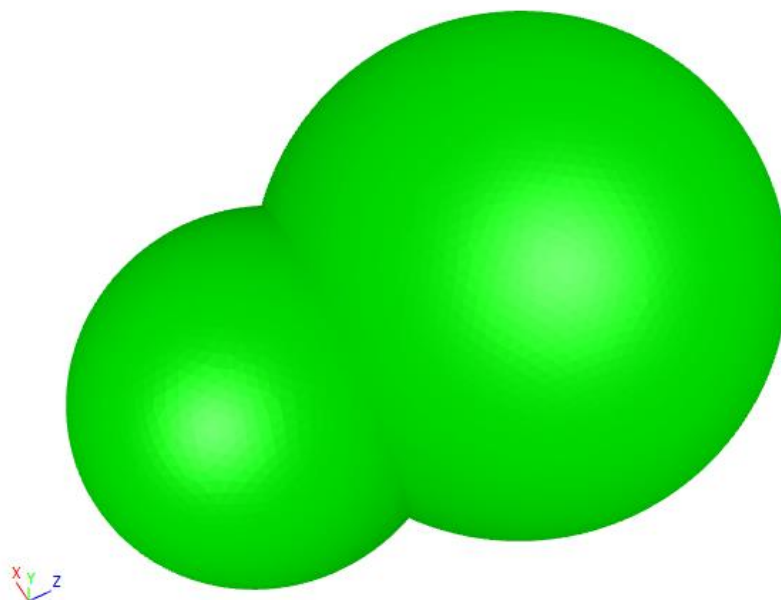


Рис. 2. Пример композиции геометрических объектов

ВСТРОЕННЫЕ ФУНКЦИИ

В языке спецификации FORTU-F имеется набор встроенных функций: элементарных математических и реализующих стандартные геометрические примитивы.

Синтаксис и семантика встроенных математических функций приведены в табл. 1.

Таблица 1 – Встроенные элементарные математические функции

№	Функция	Описание
1	2	3
1	abs(x)	Абсолютное значение аргумента x
2	acos(x)	Арккосинус
3	asin(x)	Арсинус
4	atan(x)	Арктангенс
5	atan2(x, y)	Арктангенс y/x, выраженный в радианах
6	cos(x)	Косинус
7	cosh(x)	Косинус гиперболический
8	exp(x)	Экспонента
9	sin(x)	Синус
10	sinh(x)	Синус гиперболический
11	tan(x)	Тангенс
12	tanh(x)	Тангенс гиперболический

Синтаксис и семантика встроенных функций, реализующих стандартные геометрические примитивы, приведены в табл. 2.

Таблица 2 – Встроенные геометрические примитивы

№	Функция	Описание
1	cutcone(x0,y0,z0,R0,x1,y1,z1,R1)	Усеченный конус, нижнее основание которого проходит через точку (x0,y0,z0) и имеет радиус R0, а верхнее – через (x1,y1,z1) и имеет радиус R1
2	cylinder(x0,y0,z0,x1,y1,z1,R)	Цилиндр радиуса R, основания которого проходят через точки (x0,y0,z0) и (x1,y1,z1)
3	ellipsoid(x0,y0,z0,a,b,c)	Эллипсоид с центром в точке (x0,y0,z0) и величинами полуосей a, b, и c
4	cuboid(x0,y0,z0,x1,y1,z1)	Прямоугольный параллелепипед с гранями, ортогональными осям координат, верхний левый угол которой расположен в точке (x0,y0,z0), а нижний правый – в (x1,y1,z1)
5	sphere(x0,y0,z0,R)	Сфера радиуса R с центром в точке (x0,y0,z0)
6	plane(x0,y0,z0,nx,ny,nz)	Плоскость, проходящая через точку (x0,y0,z0), и вектором нормали (nx,ny,nz)

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ

Описанный в статье язык спецификаций геометрических моделей FORTU-F был программно реализован с использованием кросс-платформенной библиотеки с открытым исходным кодом Qt [9]. На рис. 3 приведено изображение внешнего вида, реализованного

интеллектуального программного средства аналитического конструирования твердотельных геометрических моделей, получившего название QNG (Qt Net Generator).

Рассмотрим пример описания геометрической модели переходного отсека ракетносителя, представляющего собой тонкостенный усеченный конус с цилиндрическими отверстиями – технологическими люками. Эту инженерную конструкцию можно представить в виде логического вычитания из одного усеченного конуса – внешней оболочки другого усеченного конуса – внутренней стороны конструкции.

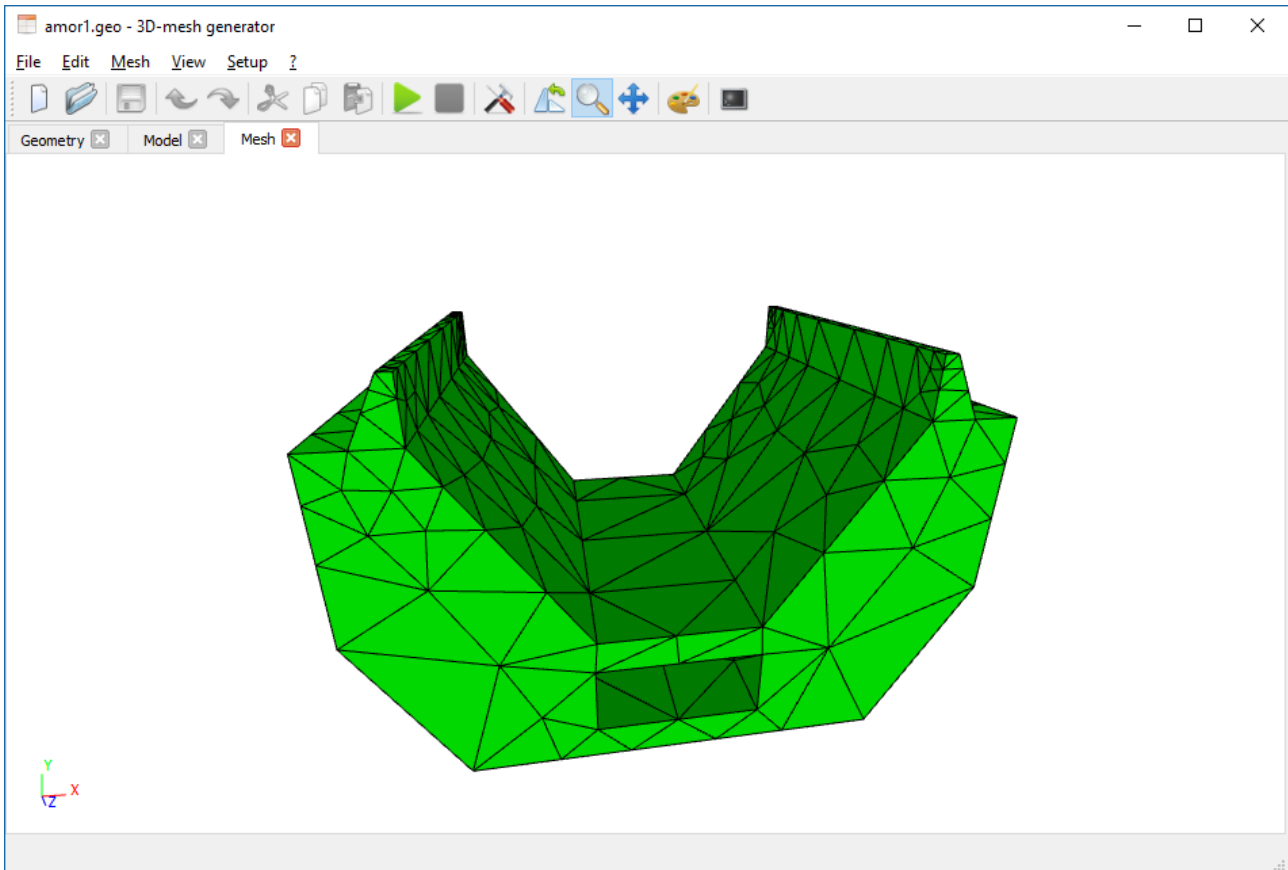


Рис. 3. Внешний вид системы QNG

Описание данной модели с помощью языка спецификаций FORTU-F приведено ниже:

!-----

! Переходный отсек ракетносителя

!-----

! Внешняя поверхность оболочки

OBJECT cone1

BEGIN

cone1 = cutcone (0, 0, 0, 3, 0, 0, 4; 4) **AND** plane (0, 0, 0, 0, 0, -1) **AND**

plane (0, 0, 4, 0, 0, 1)

END

! Внутренняя поверхность оболочки

OBJECT cone2

BEGIN

```
cone1 = cutcone (0, 0, -0.01, 2.9; 0, 0, 4.01, 3.9) AND
plane (0, 0, -0.01, 0, 0, -1) AND plane (0, 0, 4.01, 0, 0, 1)
END
```

! Отверстия в оболочке (технологические люки)

OBJECT cyl

BEGIN

```
cone1 = cylinder (4, 0, 1, -4, 0, 1, 0.5) AND plane (4, 0, 1, 1, 0, 0) AND
plane (-4, 0, 1, -1, 0, 0)
```

END

! Итоговая конструкция

OBJECT shell

BEGIN

```
shell = (cone1 AND NOT cone2) AND NOT cyl
```

END

Результат работы системы QNG – конечно-элементная дискретная модель отсека изображена на рис. 4.

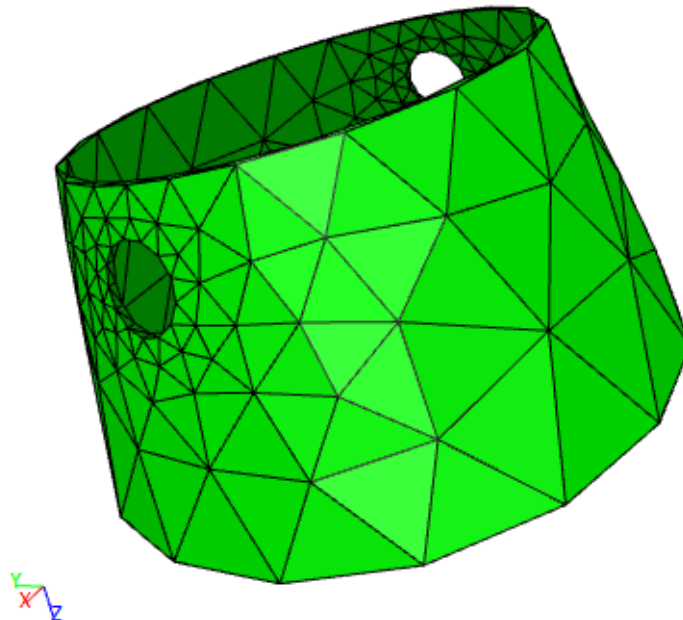


Рис. 4. Дискретная модель переходного отсека

ВЫВОДЫ

Использование функционального подхода является наиболее общим и универсальным способом построения математических моделей геометрических объектов произвольной формы. Его важным преимуществом является возможность создания библиотеки типовых геометрических объектов, которые можно повторно использовать в процессе построения моделей новых объектов произвольной сложности.

Для формализации описания математических моделей геометрических объектов с использованием функционального подхода предложен проблемно-ориентированный язык спецификаций FORTU-F, позволяющий однозначно и формально описывать функциональные математические модели двух- и трехмерных геометрических объектов произвольной формы в виде, удобном для последующей автоматической обработки с помощью компьютера. Приведены примеры описания моделей геометрических объектов на данном языке.

ЛИТЕРАТУРА

1. Bjørner D. Logics of Specification Languages / D. Bjørner, M. C. Henson. – Springer, 2007. – 646 p.
2. Голованов Н. Н. Геометрическое моделирование / Н. Н. Голованов. – М. : Издательство Физико-математической литературы, 2002, – 472 с.
3. Гоменюк С. И. Объектно-ориентированные модели и методы анализа механических процессов / С. И. Гоменюк. – Запорожье : Запорож. гос. ун-т., 2004. – 311 с.
4. Фомичёв В. С. Формальные языки, грамматики и автоматы [Электронный ресурс]. – Режим доступа : http://old.eltech.ru/misc/LGA_2007_FINAL/Allpage/content.htm.
5. Толок В. А. Метод конечных элементов: теория, алгоритмы, реализация / В. А. Толок, В. В. Киричевский, С. И. Гоменюк [и др.]. – К. : Наукова думка, 2003. – 316 с.
6. Гросс М. Теория формальных грамматик / М. Гросс, А. Лантен. – М. : Мир, 1971. – 296 с.
7. Backus J. W. The syntax and semantics of the proposed international algebraic language of the Zurich ACM-GAMM Conference / J. W. Backus. – Proc. International Conf. On Information Processing, UNESCO, 1959. – P. 125-132.
8. Feynman R. EBNF: A Notation to Describe Syntax / R. Feynman [Электронный ресурс]. – Режим доступа : <http://www.ics.uci.edu/~pattis/misc/ebnf2.pdf>.
9. Qt | Cross-platform software development for embedded & desktop [Электронный ресурс]. – Режим доступа : <http://www.qt.io/>.

REFERENCES

1. Bjørner, D. (2007), Logics of Specification Languages, Springer.
2. Golovanov, N.N. (2002), *Geometricheskoe modelirovanie* [Geometric modeling], Izdatel'stvo Fiziko-matematicheskoy literatury, Moscow.
3. Gomenyuk, S.I. (2004), *Obyektno-orientirovannyye modeli i metody analiza mexanicheskikh processov* [Object-oriented models and methods of analysis of mechanical processes], Zaporozhskiy gosudarstvenniy universitet, Zaporozhye.
4. Fomichev, V.S. (2007), “Formal languages, grammars and automata”, available at: http://old.eltech.ru/misc/LGA_2007_FINAL/Allpage/content.htm.
5. Tolok, V.A., Kirichevskiy, V.V., Gomenyuk, S.I., Grebenyuk, S.N. and Buvaylo, D.P. (2003), *Metod konechnykh elementov: teoriya, algoritmy, realizatsiya* [The Finite Element Method: Theory, Algorithms, Implementation], Naukova dumka, Kiev.
6. Gross, M. and Lanten, A. (1971), *Teoriya formal'nykh grammatik* [Theory of formal grammars], Mir, Moscow.
7. Backus, J.W. (1959), “The syntax and semantics of the proposed international algebraic language of the Zurich ACM-GAMM Conference”, *Proc. International Conf. On Information Processing, UNESCO*, pp. 125-132.
8. Feynman, R. “EBNF: A Notation to Describe Syntax”, available at: <http://www.ics.uci.edu/~pattis/misc/ebnf2.pdf>.
9. “Qt | Cross-platform software development for embedded & desktop”, available at: <https://www.qt.io/>.